COPYRIGHT NOTICE

Int. J. Electronic Security and Digital Forensics

Preserving the confidentiality of digital images using a chaotic encryption scheme

Alireza Jolfaei*

School of Information and Communication Technology,
Griffith University,
Gold Coast, QLD 4222, Australia
Email: Alireza.jolfaei@griffithuni.edu.au
*Corresponding author

Ahmadreza Matinfar and
Abdolrasoul Mirghadri

Faculty and Research Center of Communication
and Information Technology (IHU),
Tehran, Iran
Email: A_vizand@yahoo.com,
Email: Amrghdri@ihu.ac.ir

Abstract: Confidentiality of digital images is an important requirement for many multimedia applications and services. To maintain confidentiality, encryption of digital images is essential. Digital images are usually very large and encrypting such bulky data induces many performance overheads, which can be too expensive for real-time applications in resource constrained environments. In this paper, we propose a chaotic image encryption scheme which satisfies the need for both lightweightedness and security. To justify the security and efficiency, the new cipher was evaluated using a series of statistical tests. These tests included a visual testing and a histogram analysis, a randomness analysis, a correlation analysis, an entropy analysis and an image encryption quality analysis. Based on all analyses and experimental results, it is concluded that the proposed scheme is effective, efficient and trustworthy and therefore can be adopted for image encryption.

Keywords: chaos; efficiency; initialisation process; image encryption; security; stream cipher.

Reference to this paper should be made as follows: Jolfaei, A., Matinfar, A. and Mirghadri, A. (2015) 'Preserving the confidentiality of digital images using a chaotic encryption scheme', *Int. J. Electronic Security and Digital Forensics*, Vol. 7, No. 3, pp.258–277.

Biographical notes: Alireza Jolfaei is a PhD candidate in Multimedia Security at the Griffith University with research interests primarily in design and development of robust 3D content encryption schemes. His work focuses on investigating innovative solutions for maintaining the dimensional and spatial stability of encrypted content. He received a Bachelor degree (hons.) in Biomedical Engineering from Islamic Azad University, Science and Research branch, Tehran, Iran in 2007 and a Master degree (hons.) in Telecommunication Engineering from Imam Hossein Comprehensive University, Tehran, Iran in 2010.

Preserving the confidentiality of digital images

Ahmadreza Matinfar is an Assistant Professor at the Faculty and Research Center of Communication and Information Technology, IHU, Tehran, Iran. His research interests are cryptanalysis, steganography, network security, and coding theory.

Abdolrasoul Mirghadri is an Associate Professor at the Faculty and Research Center of Communication and Information Technology, IHU, Tehran, Iran. His research interests are cryptography, statistics and stochastic processes. He is the Editor-in-Chief of the *Journal of Electronic and Cyber Defence* and serves as an Associate Editor for the *Scientific Journal of Advanced Defence Science and Technology*. He is a member of ISC, ISS and IMS.

1 Introduction

1.1 Background

Over the last decade, the rapid development and diffusion of digital media and communication technologies have fundamentally altered the shape of life. Despite the obvious advancement of multimedia communications, these developments are accompanied with a number of potential security threats such as unauthorised access and breaches of privacy or confidentiality. Secure distribution of the visual data is a legitimate concern of intellectual property (IP) owners, developers, government regulatory bodies and law enforcement agencies. Therefore, there is a strong need to protect digital images against unauthorised use or other security violations. However, the confidentiality problem of digital images is beyond the simple application of established and well-known encryption methods, such as data encryption standard (DES) and advanced encryption standard (AES). This is primarily due to the constraints imposed by the data structure and the application requirements (Jolfaei et al., 2015), such as real-time performance and the security level. To address these issues, many researchers developed special encryption schemes for multimedia data. Nevertheless, most of these new schemes have been found to be insecure from the cryptanalytic point of view (Li, 2008; Guardeno, 2009; Jolfaei et al., 2014).

1.2 Related work

In Jolfaei and Mirghadri (2010b, 2010c) and Jolfaei et al. (2012a, 2012b), Jolfaei et al. investigated the application of fast stream ciphers, including A5/1 (Ekdahl and Johansson, 2003), W7 (Jolfaei and Mirghadri, 2010b) and some of the eSTREAM finalists, such as Salsa20 (Bernstein, 2008a) and HC (Wu, 2008a), for the syntax-aware image encryption. These ciphers work on binary streams. Therefore, to adapt these ciphers with the structure of 2D images, Jolfaei et al. utilised pre-processing algorithms. These studies have shown that stream ciphers are potential candidates for image encryption (Jolfaei and Mirghadri, 2011). Stream ciphers are not only appropriate for the resource constrained devices but also suitable for time critical applications which require an extreme performance in speed, memory and power consumption.

In Jolfaei and Mirghadri (2011), HC cipher (Wu, 2008a), that is one of the eSTREAM finalists, was chosen as an encryption primitive to protect digital images. The

reason behind this selection was that HC cipher stands at the top of the eSTREAM portfolio and it offers very impressive performance in software applications. HC stream cipher is considerably faster than the ten rounds 128-bit AES. On a Core 2 architecture, for example, HC-128 and HC-256 run at 2.34 and 3.66 cycles/byte for long streams, respectively, while the fastest speed reported for 128-AES is 12.59 cycles/byte (Bernstein, 2008b). In addition, there are no major, known cryptanalytic advances against the core function of the HC family of stream ciphers. However, the key initialisation process of both HC-128 and HC-256 ciphers is a weak process and is therefore vulnerable to cryptanalysis. In 2009, Liu and Qin have shown a design flaw in the initialisation process of both HC-128 and HC-256 ciphers and presented a method to recover the secret key by knowing any 16 consecutive elements in the tables. This flaw is due to the fact that the key initialisation algorithm of HC is not a one-way function. In a good key initialisation process, disclosing the master key from the expanded key stream should be difficult (Liu et al., 2006).

1.3 Contribution and organisation of the paper

To overcome the existing security flaw in the initialisation process of the HC encryption scheme, this paper improves the initialisation process by proposing a new algorithm using a Chebyshev map and a unit step function. The proposed cipher is more secure than the original scheme, which is due to the improvement of the initialisation procedure. Also, the result of statistical analyses indicates that the proposed encryption scheme has good statistical properties and can effectively encrypt digital images. The theoretical and experimental analyses confirm the effectiveness, efficiency and trustworthiness of the proposed image encryption scheme. This indicates that the proposed method is suitable for practical uses in which maintaining the confidentiality of digital images is an important requirement.

The rest of this paper is organised as follows: in Section 2, HC-128 and HC-256 stream ciphers are briefly overviewed. In Section 3, a short description of the HC image encryption scheme is given. Section 4 provides details of the proposed encryption scheme. Simulation results and security analyses are provided in Section 5. Finally, Section 6 concludes the paper.

1.4 Notation and terminology

Table 1 provides a description of symbols used throughout this article.

Table 1 Used notation

Notation	*Description*
$+$	$x + y$ means $x + y \bmod 2^{32}$, where $0 \le x < 2^{32}$ and $0 \le y < 2^{32}$
$\boxminus$	In HC-128, $x \boxminus y$ means $x - y \bmod 512$ while in HC-256, it means $x - y \bmod 1024$
$\oplus$	Bitwise XOR
$\|\|$	Concatenation
$x \gg y$	x is shifted to right by y bit-positions

Table 1 Used notation (continued)

Notation	*Description*
$x \ll y$	x is shifted to left by y bit-positions
$x \ggg n$	Means $((x \gg n) \oplus (x \ll (32 - n))$, where $0 \le n < 32$, $0 \le x < 2^{32}$
$x \lll n$	Means $((x \ll n) \oplus (x \gg (32 - n))$, where $0 \le n < 32$, $0 \le x < 2^{32}$
s_i	The keystream word generated at round i
M	A table with 512 elements of 32 bits in HC-128 while it is a table with 1,024 elements of 32 bits in HC-256
N	A table with 512 elements of 32 bits in HC-128 while it is a table with 1,024 elements of 32 bits in HC-256
K	The key is 128 bits in HC-128 while it is 256 bits in HC-256
IV	The initialisation vector is 128 bits in HC-128 while it is 256 bits in HC-256
$\lfloor N \rfloor$	The floor of the value N
P	Plain-image
C	Cipher-image
H	Number of rows in the image matrix
W	Number of columns in the image matrix
Pr (x)	Probability of observing event x
P-value	The probability of obtaining a test statistic at least as extreme as the one that was actually observed, assuming that the null hypothesis is true

2 A brief description of HC-128 and HC-256 stream ciphers

HC stream cipher is one of the successful finalists in the eSTREAM competition (Wu, 2008b). This cipher has two variants: HC-128 and HC-128. HC-128 generates a keystream from a 128-bit key (K) and a 128-bitinitialisation vector (IV). HC-128 consists of two secret tables, M and N, each one with 512 32-bit elements. At each step, one element of a table is updated with a nonlinear feedback function. All the elements of the two tables get updated every 1,024 steps. At each step, one 32-bit output is generated from the nonlinear output filtering function.

HC-256 is a word-oriented stream cipher which uses a 256-bit K and a 256-bit IV. It consists of two secret tables, each one with 1,024 32-bit elements. The tables are updated with nonlinear feedback function every 4,096 steps. The key and IV setup process and the keystream generation algorithm are described as follows.

2.1 Initialisation process (key and IV setup)

In HC-128, $K = K_0 \| K_1 \| K_2 \| K_3$ and $IV = IV_0 \| IV_1 \| IV_2 \| IV_3$ and $K_{i+4} = K_i$, $IV_{i+4} = IV_i$ for $0 \le i \le 3$, where each K_i andIV_i ($i = 0,\ldots, 3$) is a word. However, in HC-256, $K = K_0 \| K_1 \| \ldots \| K_7$ and $IV = IV_0 \| IV_1 \| \ldots \| IV_7$, where each K_i and IV_i ($i = 0,\ldots, 7$) is a word. The internal secret state of the HC algorithm consists of two tables: M and N. The following functions are used in HC-128 and HC-256 algorithms:

$$f_1(x) = (x \ggg 7) \oplus (x \ggg 18) \oplus (x \gg 3),$$
$$f_2(x) = (x \ggg 17) \oplus (x \ggg 19) \oplus (x \gg 10), \quad (1)$$

where $x = x_3 \parallel x_2 \parallel x_1 \parallel x_0$, x is a word integer that x_0 and x_3 are the least and most significant byte of x, respectively. The initialisation process of the HC cipher consists of expanding the K and IV into M and N and running the cipher 1,024 steps in HC-128 or 4,096 steps in HC-256 with the outputs being used to update M and N. This process is described as follows:

Initialisation process of HC-128

- The array W [0,…, 1,279] is obtained by expanding K and IV as follows:

$$W_i = \begin{cases} K_i & 0 \le i \le 7 \\ IV_{i-8} & 8 \le i \le 15 \\ f_2(W_{i-2}) & \\ +W_{i-7} + f_1(W_{i-15}) & 16 \le i \le 1,279 \\ +W_{i-16} + i & \end{cases} \quad (2)$$

- The tables M and N are updated using the array W as follows:

$$M[i] = W_{i+256}, \text{ for } 0 \le i \le 511, \quad N[i] = W_{i+768}, \text{ for } 0 \le i \le 511. \quad (3)$$

- The cipher is ran 1,024 steps and uses its outputs to replace the table elements.

Initialisation process of HC-256

- The array W [0,…, 2,559] is obtained by expanding K and IV as follows:

$$W_i = \begin{cases} K_i & 0 \le i \le 7 \\ IV_{i-8} & 8 \le i \le 15 \\ f_2(W_{i-2}) & \\ +W_{i-7} + f_1(W_{i-15}) & 16 \le i \le 2,559 \\ +W_{i-16} + i & \end{cases} \quad (4)$$

- The tables M and N are updated using the array W as follows:

$$M[i] = W_{i+512}, \text{ for } 0 \le i \le 1,023, \quad N[i] = W_{i+1563}, \text{ for } 0 \le i \le 1,023. \quad (5)$$

- The cipher is ran 4,096 steps and uses its outputs to replace the table elements.

2.2 The keystream generation algorithm

The following functions are used in the HC keystream generation algorithm:

Functions used in HC-128

$$g_1(x, y, z) = \big((x \ggg 10) \oplus (z \ggg 23)\big) + (y \ggg 8),$$
$$g_2(x, y, z) = \big((x \lll 10) \oplus (z \lll 23)\big) + (y \lll 8),$$
$$h_1(x) = N[x_0] + N[256 + x_2],$$
$$h_2(x) = M[x_0] + M[256 + x_2], \quad (6)$$

Functions used in HC-256

$$g_1(x, y) = \big((x \ggg 10) \oplus (y \ggg 23)\big) + N[(x \oplus y) \bmod 1,024],$$
$$g_2(x, y) = \big((x \lll 10) \oplus (y \lll 23)\big) + M[(x \oplus y) \bmod 1,024],$$
$$h_1(x) = N[x_0] + N[256 + x_1] + N[512 + x_2] + N[768 + x_3],$$
$$h_2(x) = M[x_0] + M[256 + x_1] + M[512 + x_2] + M[768 + x_3], \quad (7)$$

Using the above functions, the keystream is generated by the following procedures:

The keystream generation process of HC-128		*The keystream generation process of HC-256*	

NoIt = number of iterations that is number of required keystream words.
For $it = 1: NoIt$ **do**
$j = i \bmod 512$
IF $(i \bmod 1{,}024) < 512$ **do**
$M[j] = M[j] + g_1(M[j \boxminus 3], M[j \boxminus 10], M[j \boxminus 511])$
$s_i = h_1(M[j \boxminus 12]) \oplus M[j]$
Else
$N[j] = N[j] + g_2(N[j \boxminus 3], N[j \boxminus 10], N[j \boxminus 511])$
$s_i = h_2(N[j \boxminus 12]) \oplus N[j]$
End IF
End For
	(8)	*NoIt* = number of iterations that is number of required keystream words.
		For $it = 0: NoIt - 1$ **do**
		$j = i \bmod 1{,}024$
		IF $(i \bmod 2{,}048) < 1{,}024$ **do**
		$M[j] = M[j] + M[j \boxminus 10] + g_1(M[j \boxminus 3, M[j \boxminus 1{,}023])$
		$s_i = h_1(M[j \boxminus 12]) \oplus M[j]$
		Else
		$N[j] = N[j] + N[j \boxminus 10] + g_2(N[j \boxminus 3, N[j \boxminus 1{,}023])$
		$s_i = h_2(N[j \boxminus 12]) \oplus N[j]$
		End IF
		End For
			(9)

3 HC image encryption algorithm

To elaborate the steps of the HC image encryption algorithm, denote the plain-image by ***P*** and the cipher-image by ***C***. Note that each plain-image or cipher-image is represented by an $H \times W$ matrix, where the entry of such a matrix at position (x, y) represents a pixel. For any x $(1 \le x \le H)$ and y $(1 \le y \le W)$, let $p(x, y)$and $c(x, y)$be the pixel at the position (x, y) of the plain-image and cipher-image, respectively.

As HC is a word-oriented cipher, it generates outputs with 32 bits length. This word sequence must be converted to a byte sequence to be consistent with the pixel (byte) sequence of the image. To this end, the little-endian function described by Bernstein (2005) is employed to convert the output word sequence to a byte sequence. If the input is the word sequence $\{s(i)\}_{i=0}^{HW-1}$ then the output is the byte sequence $\{b(0+4i), b(1+4i), b(2+4i), b(3+4i)\}_{i=0}^{HW/4-1}$. The little-endian^{-1} function is described as follows:

NoIt = number of keystream words.
For $i = 0: NoIt - 1$ **do**

$$b(0+4\times i) = \bmod\left(\bmod\ \left(\bmod\left(s(i), 2^{24}\right), 2^{16}\right), 2^{8}\right)$$

$$b(1+4\times i) = \left(\bmod\left(\bmod\ \left(s(i), 2^{24}\right), 2^{16}\right) - b(0+4\times i)\right) \div 2^{8} \quad (10)$$

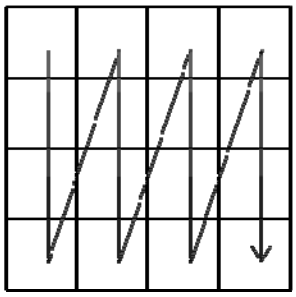
$$b(2+4\times i) = \left(\bmod\left(s(i), 2^{24}\right) - 2^{8}\times b(1+4\times i) - b(0+4\times i)\right) \div 2^{16}$$

$$b(3+4\times i) = \left(s(i) - 2^{16}\times b(2+4\times i) - 2^{8}\times b(1+4\times i) - b(0+4\times i)\right) \div 2^{24}$$

End For

Without the loss of generality, a gray-scale plain-image is denoted by a two-dimensional byte array of size $H \times W$ (height × width), $P = \{P(i, j)\}_{0\le j\le W-1}^{0\le i\le H-1}$ and the corresponding cipher-image by $C = \{C(i, j)\}_{0\le j\le W-1}^{0\le i\le H-1}$. The plain-image is considered as a 1D signal $\{P(k)\}_{k=0}^{HW-1}$ using the scanning language (Kachris, 2003). In other words, the plain-image is scanned in a column-wise raster order shown in Figure 1. The ciphertext byte sequence is generated by XOR-ing the plain-image byte sequence with the key stream byte sequence. The cipher-image is the same size as the plain-image and is constructed through column-wise raster scanning of ciphertext.

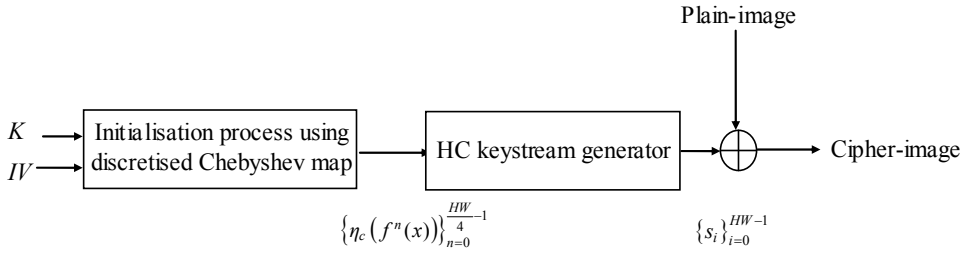
Figure 1 Column-wise raster scanning of the plain-image



4 Proposed encryption algorithm

In this section, the proposed encryption algorithm is detailed. The proposed cipher utilises a discretised Chebyshev map and a unit step function to improve the security of the HC image encryption scheme. The proposed cipher is shown in Figure 2.To elaborate the steps of the encryption algorithm, the Chebyshev map and our discretisation process are firstly detailed as follows.

Figure 2 Block diagram of the proposed image encryption algorithm



4.1 Chebyshev map

A Chebyshev map is a typical invertible iterated map that generates orthogonal real-valued sequences. Due to nonlinear properties, chaotic Chebyshev mapping has been proposed as a method of generating pseudorandom sequences (Liu, 2011; Kohda et al.,

1992). The Chebyshev map of degree D (D = 2, 3,…) is based on a trigonometric function defined as follows:

$$x_{n+1} = f(x_n) = \cos\left(D\cos^{-1}(x_n)\right), \tag{11}$$

where f: $X \to X$, $X \in [-1, 1]$. A Chebyshev map exhibits sensitive dependence on initial conditions and its invariant probability distribution density is as follows:

$$\rho(x) = \begin{cases} \left(\pi\sqrt{1-x^2}\right)^{-1} & -1 \le x \le 1, \\ 0 & \text{otherwise.} \end{cases} \tag{12}$$

The mean on chaotic sequences generated by equation (11) is

$$\bar{x} = \lim_{N\to\infty}\frac{1}{N}\sum_{i=0}^{N-1} x_i = \int_{-1}^{1} x\rho(x)dx = 0 \tag{13}$$

The normalised auto-correlation function of real-valued sequences generated by a Chebyshev map is

$$R(m) = \lim_{N\to\infty}\frac{1}{N}\sum_{i=0}^{N-1}(x_i - \bar{x})(x_{i+m} - \bar{x}) = \int_{-1}^{1} xf^m(x)\rho(x)dx - \bar{x}^2 = \begin{cases} 0.5 & m = 0, \\ 0 & m \ne 0. \end{cases} \tag{14}$$

For any two different initial values x_{01} and x_{02}, the normalised cross correlation function of the two generated sequences is:

$$\begin{aligned} C(m) &= \lim_{N\to\infty}\frac{1}{N}\sum_{i=0}^{N-1}(x_{i1} - \bar{x})(x_{(i+m)2} - \bar{x}) \\ &= \int_{-1}^{1}\int_{-1}^{1} x_1 f^m(x_2)\rho(x_1)\rho(x_2)dx_1dx_2 - \bar{x}^2 = 0. \end{aligned} \tag{15}$$

According to equations (13), (14) and (15), the Chebyshev map can generate chaotic pseudorandom real-valued sequence.

The Lyapunov exponent of the Chebyshev map, that is a quantitative measure for sensitive dependence on initial conditions, is $\ln(D)>0$ with respect to ρ. If $D \ge 2$, the Chebyshev map has a positive Lyapunov exponent for any initial state value and shows good properties with mixture and ergodicity. For D = 2, it reduces to the logistic map (Amigo et al., 2007).

4.2 Discretisation of the Chebyshev map

Data discretisation is a frequently used technique in computer science, statistics and cryptographic applications. Cryptographic algorithms are mathematical structures that are fundamentally discrete rather than continuous. Therefore, the raw real-valued sequences generated by chaotic systems could not be directly used in the cryptographic algorithms. To this end, the discretisation process makes the real-valued sequences workable in a finite precision domain. As discussed in Wah (2007), a unit step function, can be utilised to obtain binary sequences from a chaotic real-valued sequence $\{f^n(x)\}_{n=0}^{\infty}$ generated by a Chebyshev map. The unit step function is defined as

$$\eta_c(x) = \begin{cases} 0 & x < c, \\ 1 & x \geq c. \end{cases} \tag{16}$$

The threshold c is a variable equal to the median value of the chaotic real-valued sequence. For any real valued input, this function generates a binary sequence $\{\eta_c(f^n(x))\}_{n=0}^{\infty}$, namely, a Chebyshev binary sequence.

4.3 Improved HC image encryption scheme

To protect the key and *IV* setup process, a discretised Chebyshev map with a unit step function is utilised. The proposed initialisation method employs a Chebyshev binary sequence to update tables, *M* and *N*. As the secret parameters of the HC image encryption scheme, including the external key and *IV*, are binary sequences, they must be converted to real numbers before feeding the Chebyshev map. To this end, the initial value ($0 \leq x_0 \leq 1$) and the degree ($D \geq 2$) of the Chebyshev map are generated as follows:

$$K = bk_0 \| bk_1 \| \ldots \| bk_{\frac{S}{8}-1}, \tag{17}$$

$$IV = biv_0 \| biv_1 \| \ldots \| biv_{\frac{S}{8}-1}, \tag{18}$$

$$N = \sum_{i=0}^{\frac{S}{8}-1} \frac{bk_i}{256}, \tag{19}$$

$$N1 = \sum_{i=0}^{\frac{S}{8}-1} \frac{biv_i}{256}, \tag{20}$$

$$x_0 = N - \lfloor N \rfloor, \tag{21}$$

$$D = \lfloor N1 \rfloor + 2, \tag{22}$$

where each bk_i and each biv_i is a byte in the decimal equivalent form and *S* is the length of the bit sequence, that is, 128 bits for HC-128 and 256 bits for HC-256.

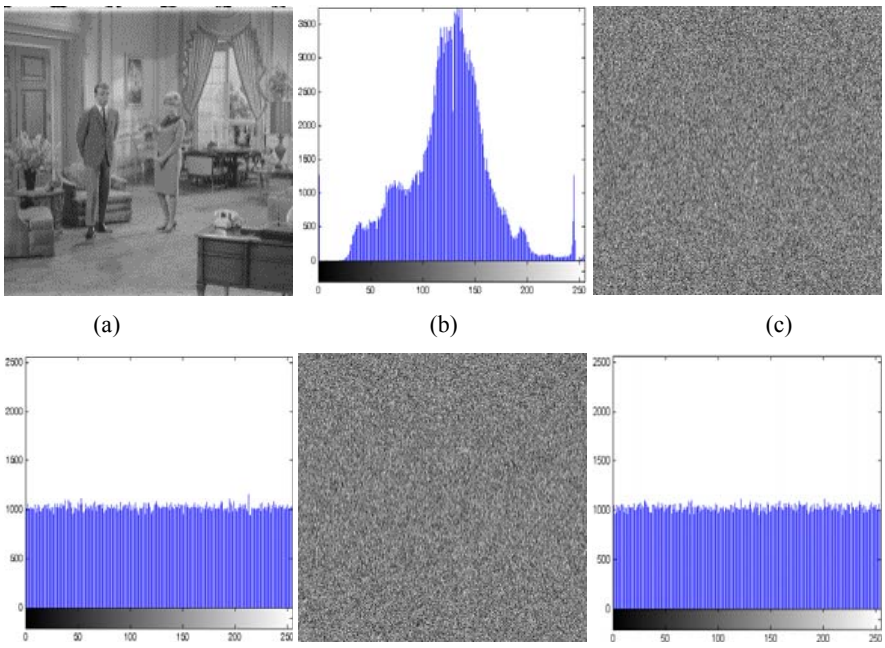
To update the tables, the Chebyshev map iterates *m* times. In HC-128, *m* is 1,024 × 32 and in HC-256, *m*is 2,048 × 32.The raw generated analogue chaotic sequence could not be directly used to update the tables. Therefore, a quantification algorithm is required to make the chaos-based random number generator workable in a discrete domain. The quantification algorithm should also guarantee pseudo-randomness and complexity of binary sequences applied to cryptography (Liu, 2011). To discretise the real-valued sequence generated by the Chebyshev map, we utilise a unit step function. Binary quantification using a unit step function has a great influence on the security and performance of the system, because the unit step function is an irreversible function. Hence, it will be difficult to derive a simple recursive equation for the recovery of the seed. As shown by Fu et al. in 2006, this method of discretisation has the following two disadvantages:

1 only one bit can be generated per iteration and therefore, the computation load is very heavy

2 increasing the number of iterations may create periodic sequences, because no processor is precision restricted.

These drawbacks can be neglected in our scheme, because the Chebyshev map iterates no more than 32,768 times in HC-256 and 65,536 in HC-256.As shown by Liu (2011), such a short sequence has good randomness properties. Thus, the chaotic binary sequences generated in the initialisation phase have the same pseudo-randomness and complexity as the chaotic real sequences that are transformed into them by the use of a quantification algorithm. By the end of the initialisation process, the HC image encryption keystream generator produces the keystream, that is $\{s_i\}_{i=0}^{HW-1}$.

Figure 3 Results of visual testing and histogram analysis using the standard test image of couple, (a) plain-image (b) plain-image histogram (c) chaotic HC-128 cipher-image (d) chaotic HC-128 cipher-image histogram (e) chaotic HC-256 cipher-image (f) chaotic HC-256 cipher-image histogram (see online version for colours)



(a) (b) (c)

(d) (e) (f)

5 Security analysis for the proposed algorithm

In this section, we have carried out several quantitative measurements to check the security and performance of the proposed cipher. We used a visual testing and a histogram analysis, a randomness analysis, a correlation analysis, an entropy analysis and an image encryption quality analysis to determine the efficiency of the new cipher and assess whether the cipher-images generated by the chaotic HC image encryption scheme

are pseudorandom and complex. A gray-scale image of size 512 × 512 (= 2,097, 152 bits), named 'couple', is selected as the test image, which is shown in Figure 3. The experiments are all performed using MATLAB 7.10 on a personal computer with a 2.0 GHz Intel dual-core processor and 2 GB RAM. We used $K = 0$ as the secret key and $IV = 0$ as the initial vector.

5.1 Visual test and histogram analysis

Visual inspection is one of the primary tests in examining the quality of encryption. A good encryption algorithm should scramble image in a way that its features are not visually distinguished. In this sub-section, a number of experimental tests were performed to evaluate the performance of the surveyed cryptosystems. The algorithms were applied on a number of gray-scale images that have the size of 512 × 512 pixels with 256 colours. Figure 3 demonstrates the visual test result and the histogram analysis for a standard test image. By comparing the original and the encrypted image in Figure 3, no meaningful visual information is observed in the encrypted image. The plain-image and the encrypted images are visually indistinguishable even with the sudden changes in the pixel intensity found in the original image.

To prevent the information leakage, it is important to ensure the statistical dissimilarity between original and encrypted images. The histogram analysis illustrates that how pixels in an image are distributed by plotting the count of observations at each colour intensity level. As shown in Figure 3, the histogram of the original image contains large sharp rises followed by sharp declines. Moreover, Figure 3 shows that the histogram of the encrypted image has a uniform distribution, which is significantly different from the histogram of the original image and has no statistical similarity in appearance. Therefore, the algorithms do not provide any clue for the statistical attacks. The histogram of the encrypted image, which is approximated by a uniform distribution, is quite different from the plain-image histogram. A relatively uniform distribution in the histogram of the cipher-image points out a good quality of the encryption method.

5.2 Randomness analysis

To ensure the security of a cryptosystem, the cipher must have some probabilistic properties, such as good distribution, long period, high complexity and efficiency. In particular, the outputs of a cryptosystem must be unpredictable in the absence of the knowledge of inputs. There are many statistical tests, each assessing the presence or absence of a pattern which, if detected, would indicate that the sequence is non-random. Recently, the NIST has designed a set of different statistical tests to examine randomness of binary sequences generated by cryptographic pseudorandom number generators. These tests verify different types of non-randomness that could exist in a sequence. The mathematical description of each test can be found at Rukhin et al. (2010). The NIST framework, like many statistical tests, is based on a hypothesis testing. A hypothesis test is a procedure for determining if an assertion about a characteristic of a population is reasonable. The output of the NIST statistical tests is a P-value which is a real number in [0, 1]. Given a significance level α, a test is interpreted as follows:

$$\begin{aligned} &P < \alpha \quad \text{the test is failed,} \\ &P \geq \alpha, \quad \text{the test is passed.} \end{aligned} \tag{23}$$

The NIST suggests using the value $\alpha = 0.01$. Thus, the confidence level of the statistical tests is 0.99. We used the NIST test suite in order to test the randomness of the surveyed algorithm. The bit sequence randomness tests were conducted on the test image shown in Figure 3. Table 2 and Table 3 present NISTSP800-22 test results of the test image. The results obtained by the NIST randomness tests illustrate that the image sequences encrypted by the encryption schemes under study have no defect and pass all the statistical tests with high *P*-values.

Table 2 SP 800-22 test result of chaotic HC-128 image encryption scheme

Tests	*P-value*	*Result*
The frequency (monobit) test	0.9362	Passed
The frequency test within a block	0.0992	Passed
The runs test	0.4426	Passed
Test for the longest run of ones in a block	0.5797	Passed
The binary matrix rank test	0.8882	Passed
Spectral test	0.6866	Passed
The non-overlapping template matching test	0.9286	Passed
The overlapping template matching test	0.8724	Passed
Maurer's 'universal statistical' test	0. 9813	Passed
The linear complexity test	0.81630	Passed
The serial test	*P*-value 1 = 0.08039, *P*-value 2 = 0.024197	Passed
The approximate entropy test	0.9647	Passed
The Cusums test	*P*-value 1 = 0.99, *P*-value 2 = 0.94379	Passed
The random excursion test		
$X = -4$	0.2787	Passed
$X = -3$	0.3956	Passed
$X = -2$	0.3675	Passed
$X = -1$	0.2572	Passed
$X = 1$	0.2302	Passed
$X = 2$	0.3569	Passed
$X = 3$	0.3432	Passed
$X = 4$	0.9808	Passed
The random excursion variant test		
$X = -9$	0.5440	Passed
$X = -8$	0.4004	Passed
$X = -7$	0.2205	Passed
$X = -6$	0.0825	Passed
$X = -5$	0.0487	Passed
$X = -4$	0.0404	Passed
$X = -3$	0.0542	Passed

Table 2 SP 800-22 test result of chaotic HC-128 image encryption scheme (continued)

Tests	*P-value*	*Result*
The random excursion variant test		
$X = -2$	0.2139	Passed
$X = -1$	0.4016	Passed
$X = 1$	0.3348	Passed
$X = 2$	0.2552	Passed
$X = 3$	0.1256	Passed
$X = 4$	0.2560	Passed
$X = 5$	0.5697	Passed
$X = 6$	0.5867	Passed
$X = 7$	0.8312	Passed
$X = 8$	0.8824	Passed
$X = 9$	0.8975	Passed

Table 3 SP 800-22 test result of chaotic HC-256 image encryption scheme

Tests	*P-value*	*Result*
The frequency (monobit) test	0.4583	Passed
The frequency test within a block	0.4640	Passed
The runs test	0.5059	Passed
Test for the longest run of ones in a block	0.3819	Passed
The binary matrix rank test	0.4882	Passed
Spectral test	0.8637	Passed
The non-overlapping template matching test	0.1273	Passed
The overlapping template matching test	0.5608	Passed
Maurer's 'universal statistical' test	0.6228	Passed
The linear complexity test	0.2385	Passed
The serial test	*P*-value 1 =0.3703, *P*-value 2 =0.26513	Passed
The approximate entropy test	0.8989	Passed
The Cusums test	*P*-value 1 = 0.87902, *P*-value 2 = 0.9236	Passed
The random excursion test		
$X = -4$	0.1956	Passed
$X = -3$	0.2966	Passed
$X = -2$	0.3266	Passed
$X = -1$	0.3525	Passed
$X = 1$	0.1302	Passed
$X = 2$	0.3132	Passed
$X = 3$	0.2592	Passed
$X = 4$	0.8205	Passed

Table 3 SP 800-22 test result of chaotic HC-256 image encryption scheme (continued)

Tests	*P-value*	*Result*
The random excursion variant test		
$X = -9$	0.2866	Passed
$X = -8$	0.2669	Passed
$X = -7$	0.2433	Passed
$X = -6$	0.2842	Passed
$X = -5$	0.4672	Passed
$X = -4$	0.7327	Passed
$X = -3$	0.9334	Passed
$X = -2$	0.3879	Passed
$X = -1$	0.1120	Passed
$X = 1$	0.7791	Passed
$X = 2$	0.8856	Passed
$X = 3$	0.3876	Passed
$X = 4$	0.2734	Passed
$X = 5$	0.4930	Passed
$X = 6$	0.8070	Passed
$X = 7$	0.8357	Passed
$X = 8$	0.8658	Passed
$X = 9$	0.9879	Passed

5.3 Correlation coefficients analysis

In an image data, each pixel is highly correlated with its adjacent pixels (Jolfaei and Mirghadri, 2011). An ideal encryption algorithm should produce cipher-images with no such correlation in the adjacent pixels. Following equations are used to study the correlation between two adjacent pixels in horizontal, vertical and diagonal orientations.

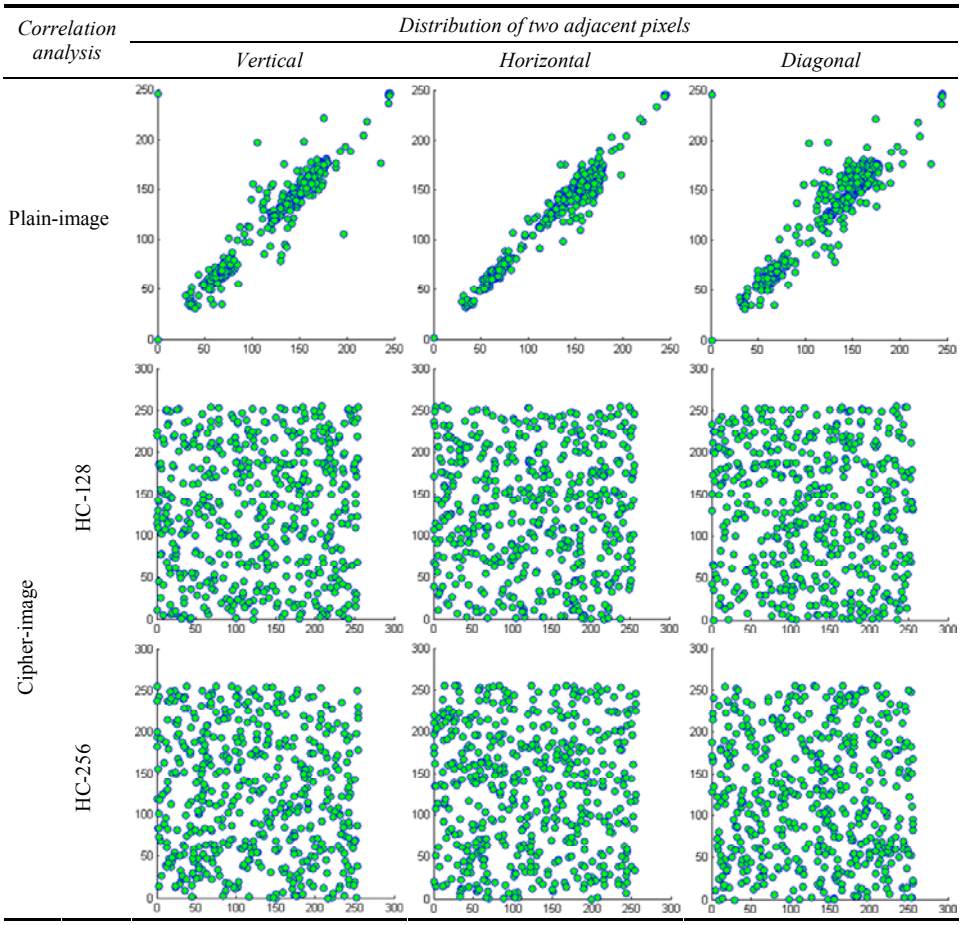
$$r_{xy} = \frac{Cov(x, y)}{\sqrt{D(x)}\sqrt{D(y)}}, \tag{24}$$

$$D(x) = \frac{1}{N}\sum_{j=1}^{N}\left(x_j - \frac{1}{N}\sum_{j=1}^{N}x_j\right)^2, \tag{25}$$

$$Cov(x, y) = \frac{1}{N}\sum_{j=1}^{N}\left(x_j - \frac{1}{N}\sum_{j=1}^{N}x_j\right)\left(y_j - \frac{1}{N}\sum_{j=1}^{N}y_j\right), \tag{26}$$

where x and y are the intensity values of two neighbouring pixels in the image and N is the number of adjacent pixels selected from the image to calculate the correlation.

Figure 4 Correlation analysis and distribution of two adjacent pixels in plain-image and cipher-image (see online version for colours)



We consider couple as the test image, which is depicted in Figure 3. Figure 4 shows the correlation distribution of two adjacent pixels in the plain-image and cipher-image. It is observed that neighbouring pixels in the plain-image are correlated too much, while there is a little correlation among neighbouring pixels in the encrypted image. Table 4 shows the results for correlation coefficients of surveyed cryptosystems. The correlation coefficients of the plain-image are far apart from the cipher-images. Results show that both variants of chaotic HC image encryption scheme have good confusion and diffusion properties and they dissipate the correlation among image pixels very well.

Table 4 Correlation coefficients of two adjacent pixels in plain-image and cipher-image

Plain-image			*Cipher-image*					
			Chaotic HC-128			*Chaotic HC-256*		
Vertical	*Horizontal*	*Diagonal*	*Vertical*	*Horizontal*	*Diagonal*	*Vertical*	*Horizontal*	*Diagonal*
0.9384	0.9798	0.9260	–0.0752	–0.0100	–0.0225	0.0686	0.0086	–0.0912

5.4 Entropy analysis

Entropy implies the amount of disorder and uncertainty in a physical system. Accordingly, the entropy of an image is an estimation of randomness and is frequently used to measure sharpness of the histogram peaks, which is directly related with better defined structural information. It is well known that the Shannon entropy $H(s)$ (Shannon, 1948) of an image s is defined as

$$H(s) = \sum_{i=0}^{255} \Pr(s_i) \log_2 \frac{1}{\Pr(s_i)} \tag{27}$$

where, s_i denotes the pixel intensity $s_i \in \{0,1,\ldots,255\}$ and $\Pr(s_i)$ represents the probability of symbol s_i. For a truly random cipher-image, the probability distribution of all pixel intensities is uniform and the maximum entropy is equal to 8. The maximum entropy is often called the ideal entropy. In practice, as the cipher-image is seldom a truly random message, the entropy value is smaller than 8. However, in good encryption systems the entropy value is very close to 8.

Table 5 Entropy value for the cipher-images of different test images

Chaotic HC-128	*Chaotic HC-256*
7.9993	7.9993

To perform an entropy analysis, a number of test images were encrypted. The number of occurrence of each gray level was recorded and the probability of occurrence was computed. The entropy test results are listed in Table 5. The values obtained are very close to the theoretical value of 8. This means that the information leakage of the encryption process is negligible and the encryption systems are secure from the entropy attack. Experiments show that both variants of the chaotic HC image ciphers have similar information entropy behaviour.

5.5 Measurement of encryption quality

Image encryption quality measure is a figure of merit employed to evaluate the effectiveness of the image encryption techniques. An encryption process transforms the pixel values in an irregular pattern. If these changes occur more irregularly, then the encryption scheme would be more effective and therefore, the encryption quality would be higher. Hence, the encryption quality may be expressed as the amount of change in pixels' values between the original image and the encrypted image (Ahmed et al., 2006), (Jolfaei and Mirghadri, 2010a). The quality of image encryption can be determined as follows (Ahmed et al., 2006):

Denote P and C by the original image and the encrypted image, respectively, each of size $H \times W$ pixels with L gray levels. Let $P(x, y)$, $C(x, y) \in \{0,\ldots L-1\}$ represent the gray levels of the images P and C at location (x, y), where $0 < x\ H - 1$, $0 < y < W - 1$. $H_L(P)$ is defined as the frequency of each gray level L in the original image (plain-image) and $H_L(C)$ as the frequency of each gray level L in the encrypted image (cipher-image). The encryption quality demonstrates the average number of changes to each gray level L and it is defined as follows:

$$\text{Encryption Quality} = \frac{\sum_{L=0}^{255} \left| H_L(C) - H_L(P) \right|}{256}. \tag{28}$$

The encryption quality was measured using the test image shown in Figure 3. The encryption quality of the Chaotic HC-128 image encryption scheme is 865.7734 while the encryption quality of Chaotic HC-256 image encryption scheme is 868.6875. This shows that HC-256 encrypts the test image more effectively.

6 Conclusions

In this paper, we proposed a chaotic image encryption scheme to address the confidentiality problem of digital images. The proposed cipher is constructed by improving the initialisation process of the HC image encryption scheme. This improvement consisted of utilising a Chebyshev map and an irreversible discretisation process. To justify the security and performance of the proposed cipher, we performed a series of statistical tests, including a visual testing, a histogram analysis, a randomness analysis, a correlation analysis, an entropy analysis and an image encryption quality analysis. The theoretical and experimental analyses demonstrate that our algorithm can effectively encrypt the plain-images. Also, compared to the original scheme, the new scheme has a more secure initialisation process. There are no statistical similarities in the appearance of the plain-images and Cipher-image and the distribution of the corresponding histogram is uniform. In addition to the good visual randomness and a uniform histogram distribution, the cipher sequence has good randomness properties. The proposed cipher induces a good confusion in the pixel values of the plain-image. In addition, according to an entropy analysis, an encryption quality assessment and a correlation analysis, both chaotic HC-128 and HC-256 image encryption schemes have qualitatively similar behaviours. Based on all analyses and experimental results, it is concluded that the proposed scheme is effective, efficient and trustworthy and therefore can be adopted for image encryption.

References

Ahmed, H.H., Kalash, H.M. and Farag Allah, O.S. (2006) 'Encryption quality analysis of RC5 block cipher algorithm for digital images', *Journal of Optical Engineering*, Vol. 45, No. 10, pp.107003–107003-7

Amigo, J.M., Kocarev, L. and Szczepanski, J. (2007) 'Theory and practice of chaotic cryptography', *Physics Letters A*, Vol. 366, No. 3, pp.211–216.

Bernstein, D.J. (2005) *Salsa20 Specification* [online] http://cr.yp.to/snuffle/spec.pdf (accessed 1 January 2015).

Bernstein, D.J. (2008a) 'The Salsa20 family of stream ciphers, the eSTREAM finalists, 4986', in Robshaw, M. and Billet, O. (Eds.): *Lecture Notes in Computer Science*, pp.84–97, Springer Berlin Heidelberg.

Bernstein, D.J. (2008b) *Which Phase-3 eSTREAM Ciphers Provide the Best Software Speeds* [online] http://cr.yp.to/streamciphers/phase3speed-20080331.pdf (accessed 1 January 2015).

Ekdahl, P. and Johansson, T. (2003) 'Another attack on A5/1', *IEEE Transactions on Information Theory*, Vol. 49, No. 1, pp.284–289.

Fu, C., Wang, P.R., Ma, X.M. and Zhu, W.Y. (2006) 'A fast pseudo stochastic sequence quantification algorithm based on Chebyshev map and its application in data encryption', *Computational Science – ICCS 2006, Lecture Notes in Computer Science*, Vol. 3991, pp.826–829.

Guardeno, D.A. (2009) *Framework for the Analysis and Design of Encryption Strategies Based on Discrete-Time Chaotic Dynamical Systems*, Doctoral Thesis, Universidad Politecticnica De Madrid.

Jolfaei, A. and Mirghadri, A. (2010a) 'A new approach to measure quality of image encryption', *International Journal of Computer and Network Security*, Vol. 2, No. 8, pp.38–44.

Jolfaei, A. and Mirghadri, A. (2010b) 'Survey: image encryption using A5/1 and W7', *Journal of Computing*, Vol. 2, No. 8, pp.1–7.

Jolfaei, A. and Mirghadri, A. (2010c) 'Survey: image encryption using Salsa20', *International Journal of Computer Science Issues*, Vol. 7, No. 5, pp.213–220.

Jolfaei, A. and Mirghadri, A. (2011) 'Substitution – permutation based image cipher using chaotic Henon and Baker's maps', *International Review on Computers and Software (IRECOS)*, Vol. 6, No. 1, pp.40–54.

Jolfaei, A., Vizandan, A. and Mirghadri, A. (2012a) 'Image encryption using HC-128 and HC-256 stream ciphers', *International Journal of Electronic Security and Digital Forensics (IJESDF)*, Vol. 4, No. 1, pp.19–42.

Jolfaei, A., Vizandan, A. and Mirghadri, A. (2012b) 'Impact of rotations in Salsa20/8 image encryption scheme', *International Journal of Computer Theory and Engineering*, Vol. 4, No. 6, pp.938–943.

Jolfaei, A., Wu, X-W. and Muthukkumarasamy, V. (2015) 'A 3D object encryption scheme which maintains dimensional and spatial stability', *IEEE Transactions on Information Forensics and Security*, Vol. 10, No. 2, pp.409–422.

Jolfaei, A., Wu, X-W. and Muthukkumarasamy, V. (2014) 'Comments on the security of diffusion-substitution based gray image encryption scheme', *Digital Signal Processing*, Vol. 32, pp.34–36.

Kachris, C. (2003) *Design and FPGA Implementation of the SCAN Encryption Algorithm*, Master Thesis, Technical University of Crete.

Kohda, T., Tsuneda, A. and Sakae, T. (1992) 'Chaotic binary sequences by Chebyshev maps and their correlation properties', *Proceedings of the IEEE Second International Symposium on Spread Spectrum Techniques and Applications (ISSSTA'92)*, Yokohama, Japan, 29 November–2 December, pp.63–66.

Li, C. (2008) *On the Security of some Multimedia Encryption Schemes*, Doctoral Thesis, City University of Hong Kong.

Liu, A. and Qin, A. (2009) 'The key and IV setup of the stream ciphers HC-256 and HC-128', *Proceedings of the 2009 International Conference on Networks Security, Wireless Communications and Trusted Computing*, pp.430–434.

Liu, N.S. (2011) 'Pseudo-randomness and complexity of binary sequences generated by the chaotic system', *Communications in Nonlinear Science and Numerical Simulation*, Vol. 16, No. 2, pp.761–768.

Liu, Y., Qin, T., Ni, W. and Zhang, S. (2006) 'Cryptanalysis of the energy efficient stream ciphers SSC2', *Secure Mobile Ad-hoc Networks and Sensors*, First International Workshop, MADNES 2005, Singapore, September 20–22, 2005, Revised Selected Papers, Vol. 4074, pp.144–157.

Rukhin, A., Soto, J., Nechvatal, J., Smid, M., Barker, E., Leigh, S., Levenson, M., Vangel, M., Banks, D., Heckert, A., Dray, J. and Vo, S. (2010) *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications* pp.800–822, NIST Special Publication 800-22, National Institute of Standards and Technology (NIST), Gaithersburg, MD [online] http://csrc.nist.gov/publications/nistpubs/800-22-rev1a/SP800-22rev1a.pdf (accessed 1 January 2015).

Shannon, C.E. (1948) 'A mathematical theory of communication', *The Bell System Technical Journal*, Vol. 27, pp.379–423, pp.623–656.

Wah, T.K. (2007) *Chaos-based Random Number Generator in Finite Precision Environment*, Master of Philosophy, City University of Hong Kong.

Wu, H. (2008a) *Cryptanalysis and Design of Stream Ciphers*, Doctoral Thesis, Katholieke Universiteit Leuven.

Wu, H. (2008b) 'The stream cipher HC-128, new stream cipher designs – the eSTREAM finalists', *Lecture Notes in Computer Science*, Vol. 4986, pp.39–47, Springer, Heidelberg, Germany.